

Reality Check

Word Problem #1

Write a portable Bash Shell script called **fileCount.bash** that uses a directory pathname as the first argument.

Your shell script will then use that argument as a positional parameter to help store all of the hidden and non-hidden file and subdirectory pathnames as positional parameters to be used to determine the total number of files contained in that directory. **Note1:** Refer to the Sample Run for exact input and output requirements... **Note2:** you can assume that when run, there will always be an argument after the command and that argument will always be a valid directory pathname...

Sample Runs (Commands or shell script appears in bold)

```
pwd
/home/msaul
./fileCount.bash . # i.e. current directory
The total number of files in "." is: 247
./fileCount.bash / # i.e. root directory
The total number of files in "/" is: 26
./fileCount.bash ./Desktop
The total number of files in "./Desktop" is: 65
```

ANSWER:

Word Problem #2

Modify your shell script **fileCount.bash** (created in Word Problem #1) to force the user to enter just one argument after the shell script. If not, the shell script must display a standard error message to the user, and terminate the shell script with a false exit status value.

Also, if the directory pathname is NOT valid, then the shell script must display a standard error message to the user, and terminate the shell script with a false exit status value.

Note1: Call this newer shell script **fileCount2.bash**

Note2: Refer to the Sample Run for exact input and output requirements...

Sample Runs (commands or shell script appear in bold)

```
pwd
/home/msaul
./fileCount2.bash      # i.e. no arguments
USAGE ./fileCount2.bash [dir-name]
echo $?
1                      # i.e. false exit status
./fileCount2.bash dir1 dir2 # i.e. 2 arguments
USAGE ./fileCount2.bash [dir-name]
echo $?
1
./fileCount2.bash moo    # i.e. one argument (invalid dir)
Error, directory pathname "moo" is not valid
echo $?
1
./fileCount2.bash .     # i.e. one argument (valid dir)
The total number of files in "." is: 248
./fileCount2.bash /
The total number of files in "/" is: 26
./fileCount2.bash ./Desktop
The total number of files in "./Desktop" is: 65
```

[Write answer on next page]

ANSWER :

Walk-thru Question #1

View the contents of the shell script below. Pretending that you are the shell interpreter, display the expected standard output. Show your rough work.

Assume that your current directory only contains the regular files **a**, **b**, **c**, and the shell script below which is called **walk1.bash**. Assume that the shell script *walk1.bash* has appropriate permissions to run for that user.

Contents of shell script **walk1.bash**:

```
#!/bin/bash
echo '$*'
echo "$#"
if [ -f a ]
then
    echo -n "Yippee!"
fi
if [ ! -f b ]
then
    echo "Horray"
fi
if [ ! -d c ]
then
    echo "Boo!"
fi
echo "Done!"
```

OUTPUT

ROUGH WORK